AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for transparently optimizing data access, comprising:
gathering information related to data usage when a system is processing using a client runtime; and
determining a usage pattern of the system using gathered information.

2. (Currently Amended) The method of claim 1, further comprising:
pre-fetching data determined by the usage pattern of the system;
caching data locally in a cache associated with the system;
accessing data from the cache ~~by the system~~; and
synchronizing cached data with persistent data.

3. (Original) The method of claim 1, further comprising:
generating a description of a business process model by retaining the usage pattern over a period of at least one execution of the system.

4. (Original) The method of claim 1, further comprising:
generating tests for the system by retaining the usage pattern over a period of at least one execution of the system.

5. (Original) The method of claim 1, wherein the usage pattern comprises pieces of information used together based on a relationship.

6. (Original) The method of claim 5, wherein the relationship is temporal.

7. (Original) The method of claim 5, wherein the relationship is causal.

8. (Original) The method of claim 1, further comprising:
deriving an initial usage pattern from application code analysis.

9. (Original) The method of claim 1, further comprising:
deriving an initial usage pattern from an empty set.

10. (Currently Amended) The method of claim 1, further comprising:
deriving an initial usage pattern from a specification of the system.

11. (Original) The method of claim 1, further comprising:
displaying the usage pattern to a display device.

12. (Original) The method of claim 1, further comprising:
generating documentation from the usage pattern.

13. (Currently Amended) A method for transparently optimizing data access, comprising:
gathering information related to data usage when a system is processing using a client runtime;
determining a usage pattern of the system using gathered information;
pre-fetching data determined by the usage pattern of the system;
caching data locally in a cache associated with the system;
accessing data from the cache ~~by the system~~; and
synchronizing cached data with persistent data.

14. (Currently Amended) A method for transparently optimizing a distributed application having a client-side and a server-side, comprising:
gathering information related to data usage on the client-side when the distributed application is processing using a client runtime; and
determining a usage pattern using gathered information.

15. (Currently Amended) The method of claim 14, further comprising:
pre-fetching data from the server-side using the usage pattern and a server runtime;
caching data on the client-side in a cache associated with the client-side;
accessing data on the client-side using the cache; and
synchronizing cached data on the client-side with persistent data on the server-side.

16. (Original) The method of claim 14, wherein the usage pattern comprises pieces of information used together based on a relationship.

17. (Original) The method of claim 16, wherein the relationship is temporal.

18. (Original) The method of claim 16, wherein the relationship is causal.

19. (Original) The method of claim 14, wherein the data represents objects.

20. (Original) The method of claim 14, further comprising:
deriving an initial usage pattern from application code analysis.

21. (Original) The method of claim 14, further comprising:
deriving an initial usage pattern from an empty set.

22. (Currently Amended) The method of claim 14, further comprising:
deriving an initial usage pattern from a specification of the system.

23. (Original) The method of claim 14, further comprising:
displaying the usage pattern to a display device.

24. (Original) The method of claim 14, further comprising:
generating documentation from the usage pattern.

25. (Currently Amended) A method for transparently optimizing a distributed application having a client-side and a server-side, comprising:
gathering information related to data usage on the client-side when the distributed application is processing using a client runtime;
determining a usage pattern using gathered information;
pre-fetching data from the server-side using the usage pattern and a server runtime;
caching data on the client-side in a cache associated with the client-side;
accessing data on the client-side using the cache; and
synchronizing cached data on the client-side with persistent data on the server-side.

26. (Currently Amended) A computer-readable medium having recorded thereon instructions executable by processing, the instructions for:
gathering information related to data usage when a system is processing using a client runtime;
determining a usage pattern of the system using gathered information;

pre-fetching data determined by the usage pattern of the system;

caching data locally in a cache associated with the system;

accessing data from the cache ~~by the system~~; and

synchronizing cached data with persistent data.

27. (Currently Amended) An apparatus for transparently optimizing data access, comprising:

means for gathering information related to data usage when a system is processing using a client runtime;

means for determining a usage pattern of the system using gathered information;

means for pre-fetching data determined by the usage pattern of the system;

means for caching data locally in a cache associated with the system;

means for accessing data from the cache ~~by the system~~; and

means for synchronizing cached data with persistent data.